

### REMARKS

In this Response, Applicant cancels claims 6, 7, 16, and 17, amends claims 1-3, 9, 11-13, 19, and 22-24, and removes the basis for the Examiner's rejections. Cancellations of and amendments to the claims are being made solely to expedite prosecution of the present application and do not constitute an acquiescence to any of the Examiner's rejections. Support for the amendments to the claims can be found throughout the application. Applicant reserves the option to further prosecute the same or similar claims in the present or a subsequent application. Upon entry of the Amendment, claims 1-5, 8-15, and 18-24 are pending in the present application.

#### Claim Rejections

The Examiner rejected claims 1-24 under 35 U.S.C. § 103(a) as being unpatentable over Hardwick in view of Ball.

#### *Claims 1-10*

Applicant's independent claim 1 is directed to a method for providing a self-tuning object to a user program. Among other things, Applicant's independent claim 1 includes detecting expressions that include the self-tuning object during a simulated execution of the user program, generating a trace file including data that represents the expressions and the sequence in which the expressions occur in the user program and that enables generation of source code corresponding to the expressions, dividing the trace file into trace file blocks, converting the trace file blocks into source code expression blocks, generating minimal timing, compiled expression blocks for the source code expression blocks, and linking the minimal timing source code expression blocks into the user program. *The minimal timing, compiled expression blocks are generated on the basis of parameterizing the source code expression blocks to include loop blocking and/or loop unrolling parameters and identifying the values of the parameters that are associated with minimal execution times for the compiled expression blocks.*

Hardwick describes a system for converting a program that is written in a nested parallel language into a sequential programming language and calls to a message parsing interface. Most relevantly, Hardwick receives a user program written in the nested parallel language, identifies in the user program recursive operations on vector objects capable of being executed in parallel, computes the computational cost for each of the recursive operations, and assigns, on the basis of

20/573816.2

divide-and-conquer algorithms, each of the recursive operations to a number of processors that minimizes the computational cost of the recursive operation. (Hardwick col. 4, l. 55 to col. 5, l. 7, col. 6, ll. 4-29, col. 8, l. 33 to col. 9, l. 16, col. 29, l. 33 ff, Hardwick claim 16, and Hardwick Figs. 5A-5D.) Although Hardwick minimizes the computational cost of recursive operations on vector objects, Hardwick does not minimize the computational cost of recursive operations on vector objects by parameterizing the source code for the recursive operations to include loop blocking and/or loop unrolling parameters and identifying the values of the parameters that result in minimal timing for executing the compiled source code. Rather, Hardwick minimizes the computational cost of a recursive operation by determining the optimum number of processors to which to assign the recursive operation.

In contrast to Hardwick, Applicant's independent claim 1 describes generating minimal timing, compiled expression blocks on the basis of *parameterizing source code expression blocks to include loop blocking and/or loop unrolling parameters and identifying the values of the parameters that are associated with minimal timing for the compiled expression blocks*. Hardwick does not, therefore, teach or suggest at least the feature of Applicant's independent claim 1 directed to generating minimal timing, compiled expression blocks on the basis of *parameterizing source code expression blocks to include loop blocking and/or loop unrolling parameters and identifying the values of the parameters that are associated with minimal execution time for the compiled expression blocks*.

As for Ball, Ball describes a system for predicting the performance of a processor and does not contain any teaching or suggestion related to generating minimal timing, compiled expression blocks. Ball does not, therefore, teach or suggest at least the feature of Applicant's independent claim 1 directed to generating minimal timing, compiled expression blocks on the basis of *parameterizing source code expression blocks to include loop blocking and/or loop unrolling parameters and identifying the values of the parameters that are associated with minimal execution time for the compiled expression blocks*.

Independent claim 1 is therefore allowable. Since claims 2-5 and 8-10 depend from independent claim 1, claims 2-5 and 8-10 are also allowable.

*Claims 11-24*

Applicant's independent claims 11 and 22-24 are directed to computer program products, computer data signals, and systems that include features similar to independent method claim 1. Applicant's independent claims 11 and 22-24 are therefore allowable for the reasons provided with respect to independent claim 1. Since claims 12-15 and 18-24 depend from independent claim 11, claims 12-15 and 18-24 are also allowable.

**CONCLUSION**

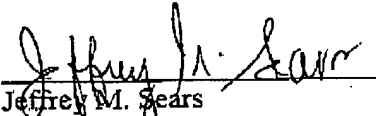
This Response is fully responsive to the present Office Action.

Based on the foregoing Amendment and Remarks, Applicant respectfully submits that this application is in condition for allowance. Accordingly, Applicant requests allowance. Applicant invites the Examiner to contact the Applicant's undersigned Attorney if any issues are deemed to remain prior to allowance.

Respectfully submitted,  
FOLEY HOAG LLP

Date: 1/29/04

Customer No. 25,181  
Patent Group  
Foley Hoag LLP  
155 Seaport Blvd.  
Boston, MA 02210  
Tel: (617) 832-3022  
Fax: (617) 832-7000

  
Jeffrey M. Sears  
Reg. No. 48,440  
Attorney for the Applicant